DATA MINING — Comprehensive Exam Answers

BITS Pilani | WILP Division | Course: ZC425

Question 1 — Short Answers (5 × 2 marks)

(a) Cluster Distance — Single Link and Complete Link

Clusters: C1 = {2, 4, 10}, C2 = {20, 30, 311}

All pairwise distances between C1 and C2:

Pair	Distance
2 & 20	18
2 & 30	28
2 & 311	309
4 & 20	16
4 & 30	26
4 & 311	307
10 & 20	10
10 & 30	20
10 & 311	301

Single-Link (MIN): Minimum distance = **10** (pair: 10 & 20)

Complete-Link (MAX): Maximum distance = **309** (pair: 2 & 311)

(b) Pattern in Sales Data

Yes — a clear seasonal/temporal pattern is visible *without* any algorithmic data mining:

- Sales drop to **100** in mid-December (15-12-2016) suggesting a seasonal low.
- Sales spike to **500–600** in April–May (festival/summer season) and again dip mid-year.
- Most other dates show a baseline of **200**, with peaks in April–May.

Pattern: Sales follow a seasonal cycle — peak around April–May, trough in December. This is directly observable by sorting the data by date and inspecting values.

(c) Cosine Similarity and Euclidean Distance for x=(4,4,4,4), y=(2,2,2,2)

Cosine Similarity:

$$x \cdot y = 4\times2 + 4\times2 + 4\times2 + 4\times2 = 8+8+8+8 = 32$$

$$|x| = \sqrt{(16+16+16+16)} = \sqrt{64} = 8$$

$$|y| = \sqrt{(4+4+4+4)} = \sqrt{16} = 4$$

$$\cos(x,y) = 32 / (8 \times 4) = 32/32 = 1.0$$

Cosine similarity = **1.0** (vectors point in the same direction — perfectly similar).

Euclidean Distance:

$$d = \sqrt{[(4-2)^2 + (4-2)^2 + (4-2)^2 + (4-2)^2]}$$

```
= √[4+4+4+4] = √16 = 4
```

Euclidean distance = **4**

(d) kNN Classification of x = 5.0

Distances from x=5.0 to all points (sorted):

Distance	Point	Label
0.1	4.9	+
0.2	5.2	–
0.4	4.6	+
0.5	4.5	+
0.6	5.6	–
0.8	5.8	+
1.8	3.2	–
2.1	7.1	–
4.4	0.6	–
4.5	9.5	–

3-NN: 3 nearest = 4.9(+), 5.2(–), 4.6(+) → 2× '+', 1× '–' → **Classified as '+'**

9-NN: 9 nearest (exclude 9.5) = 4.9(+),5.2(–),4.6(+),5.6(–),4.5(+),5.8(+),3.2(–),7.1(–),0.6(–) → 4× '+', 5× '–' → **Classified as '–'**

(e) Bread ⇒ Cheese or Cheese ⇒ Bread ?

Confidence of a rule X ⇒ Y = support(X ∪ Y) / support(X).

Since **bread sells more often than cheese**, support(bread) > support(cheese).

Therefore: confidence(cheese ⇒ bread) = support(bread ∩ cheese) / support(cheese) will be **higher** than confidence(bread ⇒ cheese) = support(bread ∩ cheese) / support(bread).

Conclusion: We should prefer **cheese ⇒ bread** because it has higher confidence (smaller denominator), making it a stronger association rule.

Question 2 — Concise Responses (6 × 3 marks)

(a) Binary Attribute Distances

Object1: (1,1,0,0), Object2: (1,0,1,0)

Contingency table: f11=1, f10=1, f01=1, f00=1 (11=both 1; 10=O1=1,O2=0; 01=O1=0,O2=1; 00=both 0)

(a) Symmetric distance (Simple Matching):

```
d = (f10 + f01) / (f11 + f10 + f01 + f00) = (1+1)/(1+1+1+1) = 2/4 = 0.5
```

(b) Asymmetric distance (0-0 matches ignored):

```
d = (f10 + f01) / (f11 + f10 + f01) = (1+1)/(1+1+1) = 2/3 ≈ 0.667
```

(c) Jaccard Coefficient (similarity, not distance):

```
J = f11 / (f11 + f10 + f01) = 1/(1+1+1) = 1/3 ≈ 0.333
```

Jaccard distance = 1 – 1/3 = **2/3** ≈ **0.667**

(b) Binning: 5, 10, 11, 13, 15, 35, 50, 55, 72, 90, 204, 215

(a) Equal-frequency (4 values per bin):

```
Bin 1: 5, 10, 11, 13
Bin 2: 15, 35, 50, 55
Bin 3: 72, 90, 204, 215
```

(b) Equal-width (range = 215–5 = 210; width = 70):

```
Bin 1 [5–75): 5, 10, 11, 13, 15, 35, 50, 55, 72
Bin 2 [75–145): 90
Bin 3 [145–215]: 204, 215
```

(c) Clustering: Using natural groupings visible in the data:

```
Bin 1 (small): 5, 10, 11, 13, 15
Bin 2 (medium): 35, 50, 55, 72, 90
Bin 3 (large): 204, 215
```

(c) Fraud Detection and Data Mining

Fraud detection leverages several DM techniques:

Classification: Train models (Decision Trees, Neural Networks, SVM) on labelled fraud/non-fraud transactions to predict future fraud.

Clustering: Identify outlier groups or unusual spending patterns that deviate from normal behaviour clusters.

Association Rules: Discover co-occurring suspicious activities (e.g., multiple transactions from different locations within minutes).

Anomaly/Outlier Detection: Flag transactions that fall far outside a customer's historical profile. Real-time stream mining enables instant alerts.

(d) Bagging vs Boosting

Aspect	Bagging	Boosting
Idea	Parallel ensemble of independent learners	Sequential ensemble; each learner corrects predecessor
Sampling	Bootstrap (random w/ replacement)	Weighted sampling (misclassified get higher weight)
Combination	Majority vote / average	Weighted majority vote
Bias/Variance	Reduces variance	Reduces bias
Overfitting	Less prone	Can overfit noisy data
Example	Random Forest	AdaBoost, Gradient Boosting

(e) Multimedia vs Business Data Mining

Aspect	Business Data Mining	Multimedia Data Mining
Data Type	Structured (tables, transactions)	Unstructured (images, audio, video, text)

Representation	Numbers, categories	Feature vectors, colour histograms, MFCC
Complexity	Simpler	Higher; requires feature extraction first
Tasks	Market basket, fraud, churn	Content-based retrieval, scene classification
Tools	SQL, OLAP	Computer vision, signal processing + ML

(f) Candidates generated by Apriori from a frequent 50-itemset

Apriori generates (k+1)-itemset candidates from frequent k-itemsets by joining pairs that share the first (k–1) items.

From a single frequent 50-itemset, joining it with itself produces candidates by adding one item. The number of candidates = **C(50,2) + C(50,1)** is not the right framing.

More precisely: from one frequent 50-itemset there is exactly **1** candidate 51-itemset (the set itself extended — but there's no other 50-itemset to join with).

If interpreted as: 'database has exactly one frequent 50-itemset', then Apriori generates **0 candidates** for size 51 (need two frequent 50-itemsets sharing first 49 items to join). Minimum candidates = **0** for the 51-itemset level; the 50-itemset itself was 1 candidate at its level.

Question 3 — FP-Growth & Association Rules (5+2 marks)

Step 1 — Frequent 1-itemsets (min_sup = 60% = 3/5 transactions)

Item	Count	Frequent?
Butter	5	Yes
Bread	4	Yes
Beans	3	Yes
Jam	3	Yes
Milk	3	Yes
Potato	2	No
Apple	2	No
Shampoo	1	No
Soap	1	No
Onion	1	No
Banana	1	No

Frequent items (support ≥ 3): **Butter(5), Bread(4), Beans(3), Jam(3), Milk(3)**

Order by frequency: Butter > Bread > Beans > Jam > Milk

Step 2 — Reordered transactions (only frequent items, sorted by frequency)

TID	Items (ordered)
T100	Butter, Bread, Beans, Jam, Milk
T200	Butter, Bread, Jam, Milk
T300	Butter, Bread, Beans
T400	Butter, Beans, Milk

T500	Butter, Bread, Jam

Step 3 — FP-Tree Construction & Mining: All Frequent Itemsets

After building the FP-tree and mining conditional pattern bases, all frequent itemsets are:

Frequent Itemset	Support	Support %
{Butter}	5	100%
{Bread}	4	80%
{Beans}	3	60%
{Jam}	3	60%
{Milk}	3	60%
{Butter, Bread}	4	80%
{Butter, Beans}	3	60%
{Butter, Jam}	3	60%
{Butter, Milk}	3	60%
{Bread, Beans}	3	60%
{Bread, Jam}	3	60%
{Butter, Bread, Beans}	3	60%
{Butter, Bread, Jam}	3	60%

Step 4 — Strong Association Rules: buys(X,i1) ∧ buys(X,i2) ⇒ buys(X,i3) [min_conf=80%]

We need 3-itemset rules. Frequent 3-itemsets: {Butter,Bread,Beans} sup=3/5=60%, {Butter,Bread,Jam} sup=3/5=60%

Rule	Confidence	Valid?
buys(X,Butter) ∧ buys(X,Bread) ⇒ buys(X,Beans)	3/4 = 75%	■ < 80%
buys(X,Butter) ∧ buys(X,Beans) ⇒ buys(X,Bread)	3/3 = 100%	✓ Strong
buys(X,Bread) ∧ buys(X,Beans) ⇒ buys(X,Butter)	3/3 = 100%	✓ Strong
buys(X,Butter) ∧ buys(X,Bread) ⇒ buys(X,Jam)	3/4 = 75%	■ < 80%
buys(X,Butter) ∧ buys(X,Jam) ⇒ buys(X,Bread)	3/3 = 100%	✓ Strong
buys(X,Bread) ∧ buys(X,Jam) ⇒ buys(X,Butter)	3/3 = 100%	✓ Strong

Strong rules (conf ≥ 80%, sup ≥ 60%):

buys(X, Butter) ∧ buys(X, Beans) ⇒ buys(X, Bread) [s=60%, c=100%]

buys(X, Bread) ∧ buys(X, Beans) ⇒ buys(X, Butter) [s=60%, c=100%]

buys(X, Butter) ∧ buys(X, Jam) ⇒ buys(X, Bread) [s=60%, c=100%]

buys(X, Bread) ∧ buys(X, Jam) ⇒ buys(X, Butter) [s=60%, c=100%]

Question 4 — Linear Regression (3+2 marks)

(a) Least Squares Regression: Price = a + b × Area

Given data:

Area (X)	Price (Y)	X^2	XY

72	84	5184	6048
50	63	2500	3150
81	77	6561	6237
74	78	5476	5772
94	90	8836	8460
86	75	7396	6450
ΣX=457	**ΣY=467**	**ΣX²=35953**	**ΣXY=36117**

Using the least squares formulas:

$$n=6,\ \Sigma X=457,\ \Sigma Y=467,\ \Sigma X^2=35953,\ \Sigma XY=36117$$

$$b = (n\Sigma XY - \Sigma X\Sigma Y) / (n\Sigma X^2 - (\Sigma X)^2)$$

$$= (6\times36117 - 457\times467) / (6\times35953 - 457^2)$$

$$= (216702 - 213419) / (215718 - 208849)$$

$$= 3283 / 6869 = 0.4779$$

$$a = (\Sigma Y - b\times\Sigma X) / n = (467 - 0.4779\times457) / 6 = 41.4299$$

Regression Equation: Price = 41.43 + 0.4779 × Area

(b) Predict price for Area = 80 sq. m

$$\text{Price} = 41.43 + 0.4779 \times 80 = 41.43 + 38.24 = 79.67$$

Predicted Price ≈ ■79.67 lakh

Question 5 — K-Means Clustering (K=2) (5 marks)

Objects: 11 points with attributes A and B.

Initial Centroids (chosen as first two points): C1 = Object1(1,1), C2 = Object2(1.5,2)

Note: A common choice is to take the 2 most spread-out points. Using C1=(1,1) and C2=(7,10) for better convergence.

Initial Centroids: C1=(1,1), C2=(7,10)

Iteration 1:

Obj	A	B	d to C1(1.00,1.00)	d to C2(7.00,10.00)	Cluster
O1	1	1	0.00	10.82	C1
O2	1.5	2	1.12	9.71	C1
O3	3	4	3.61	7.21	C1
O4	3.5	5	4.72	6.10	C1
O5	4	4.5	4.61	6.26	C1
O6	7	10	10.82	0.00	C2
O7	5.5	5	6.02	5.22	C2
O8	6	8.5	9.01	1.80	C2
O9	6.5	8	8.90	2.06	C2
O10	7	7.5	8.85	2.50	C2

O11	5	7	7.21	3.61	C2

New C1 = (2.600, 3.300), New C2 = (6.167, 7.667)

Iteration 2:

Obj	A	B	d to C1(2.60,3.30)	d to C2(6.17,7.67)	Cluster
O1	1	1	2.80	8.43	C1
O2	1.5	2	1.70	7.34	C1
O3	3	4	0.81	4.84	C1
O4	3.5	5	1.92	3.77	C1
O5	4	4.5	1.84	3.84	C1
O6	7	10	8.02	2.48	C2
O7	5.5	5	3.36	2.75	C2
O8	6	8.5	6.21	0.85	C2
O9	6.5	8	6.11	0.47	C2
O10	7	7.5	6.08	0.85	C2
O11	5	7	4.41	1.34	C2

New C1 = (2.600, 3.300), New C2 = (6.167, 7.667)

Centroids unchanged — Algorithm converged!

Final Clusters:

Cluster 1 (centroid≈2.60,3.30): O1(1, 1), O2(1.5, 2), O3(3, 4), O4(3.5, 5), O5(4, 4.5)

Cluster 2 (centroid≈6.17,7.67): O6(7, 10), O7(5.5, 5), O8(6, 8.5), O9(6.5, 8), O10(7, 7.5), O11(5, 7)

Question 6 — F-Score for Cluster Quality (2+3 marks)

(a) How F-Score Quantifies Cluster Quality

F-Score combines **Precision** and **Recall** for a cluster-class pair:

```
Precision(i,j) = n_ij / n_i [fraction of cluster i that belongs to class j]
Recall(i,j) = n_ij / n_j [fraction of class j captured by cluster i]
F(i,j) = 2 × P(i,j) × R(i,j) / (P(i,j) + R(i,j))
F-Score = Σ_j (n_j/n) × max_i F(i,j)
```

A higher F-Score (closer to 1) indicates better clustering quality — clusters align well with true classes.

(b) Computing F-Score for Metro and Financial classes

Given table:

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Total
#1	1	1	5	11	4	676	698
#2	27	89	333	827	253	33	1562
#3	126	465	8	105	16	29	749
Total	**154**	**555**	**346**	**943**	**273**	**738**	**3009**

For METRO class (n_Metro = 943, n_total = 3009):

```
C1: P=11/698=0.0158, R=11/943=0.0117, F=0.0134
C2: P=827/1562=0.5294, R=827/943=0.8770, F=0.6603
C3: P=105/749=0.1402, R=105/943=0.1113, F=0.1241
```

Best F for Metro = max over clusters = **F(C2, Metro)**

```
F(C2,Metro) = 2 × 0.5294 × 0.8770 / (0.5294+0.8770) = 0.6603
```

For FINANCIAL class (n_Financial = 555, n_total = 3009):

```
C1: P=1/698=0.0014, R=1/555=0.0018, F=0.0016
C2: P=89/1562=0.0570, R=89/555=0.1604, F=0.0841
C3: P=465/749=0.6208, R=465/555=0.8378, F=0.7132
```

Best F for Financial = **F(C3, Financial)** = 0.7132

Overall F-Score (weighted, for Metro and Financial only):

```
F = (943/3009)×0.6603 + (555/3009)×0.7132
= 0.3134×0.6603 + 0.1844×0.7132 = 0.3385
```

Overall F-Score (Metro + Financial) ≈ **0.3385**

Assumptions:

- Each document belongs to exactly one true class.
- Each document belongs to exactly one cluster.
- We consider only the Metro and Financial classes as requested.